B. O'Neill
Emmanuel College
bo271

Computer Science Tripos Part II Project Proposal

Efficient Asymmetric Cryptography for RFID Access Control

11 October 2017

Project Originator: Dr M. Kuhn

Resources Required: See attached Project Resource Form

Project Supervisor: Dr M. Kuhn

Signature:

Director of Studies: Dr T. Sauerwald

Signature:

Overseers: Dr R. Mantiuk and Dr I. Wassell

Signatures:

Introduction

The university, along with many other institutions and systems, uses smartcards that use the MIFARE classic chip.

In a presentation by Henryk Plötz and Karsten Nohl at the Chaos Communication Congress in 2007, they described how the security of the MIFARE classic chip relied largely on obfuscation of the implementation details. This flaunts Kerchoff's principle, which states that encryption security should rely solely on the key and that not on the secrecy of the system. They reverse engineered the MIFARE classic chip and discovered various weaknesses with the chip which won't be discussed at length here.

Many compromises were made on the security of the tag as a result of the very limited computing power of the chips, but others were unnecessary, and advances in available computing power should allow for better cryptographic methods.

One more general weakness of the MIFARE chip, in common with most other RFID tags, is the use of symmetric cryptography. When used in a large organisation such as the University to enforce door access control (among other things), limited on-chip memory means there are typically only a small number of private keys stored in most cards that are used to open most doors. If any card containing some common key is compromised, that card can open any door responding to that key until either the card is reprogrammed, or the key for the doors are changed at great inconvenience to all users.

The successor to the MIFARE classic, MIFARE plus, solves many of the problems with the classic chip, but still uses symmetric cryptography so does not solve the stolen key problem.

A University working group is currently exploring a successor system to the MIFARE Classic system, and a successful implementation of a prototype system could help usefully inform this process.

Resources Required

- At least two programmable Java Cards. Multiple cards would be useful to save switching between card profiles when testing, for testing potential security vulnerabilities involving two cards, and in case one is lost or broken. Java Cards cost no more than a few pounds each online.
- Java Card development software. Can use free NetBeans Java Card plugin.
- Contactless ISO14443-compatible card reader. Available on Amazon for £39.
- Privately owned PC. Remote backups will be made using GitHub.
- (Potentially) Raspberry Pi to run the door controller.

Starting Point

Basic knowledge of cryptography from the Part 1B course Security I, and general programming experience.

Substance and Structure of the Project

The objective of the project will be to develop a solution to the danger of compromised smartcards described in the introduction, where each door controller in a large organisation can have different access configurations, and compromised smartcards pose a minimal security risk.

This will be done using asymmetric cryptography such as digital signature algorithms, where each card will hold one, or possibly more, unique private keys, and and door controllers will not contain any secrets.

There are multiple protocols in existence that may suit this purpose, such as NIST 800-73, OPACITY, and PLAID. These will be compared, and one will be implemented. The primary metrics by which they will be compared will relate to speed of execution and security, but other metrics such as scalability and adaptability to possible future modifications will also be considered.

Other weaknesses in the security of the MIFARE classic RFID chip will also be looked at, and solutions will be considered for the new implementation, resulting in a more secure and practical alternative.

A full implementation of the prototype system will include at least the following:

- A Java Card app.
- Card personalization software running on a PC for configuring the card.
- Door controller application that verifies the card. It should also have a suitable management interface.

Optional Extensions

If work progresses at the expected rate, there should be time to incorporate some additional features. Possibilities include:

- Implement more than one of the protocols and directly compare their performance and security.

- Either make the card backwards compatible with MIFARE classic, or propose some other smooth rollout strategy, so that the new system could seamlessly replace the old university card system.
- Experiment with the timing predictability of the smartcard used, implement a distance-bounding protocol for the asymmetric access control system to prevent relay attacks or other remote attacks.
- Implement a (partial) MIFARE Plus or MIFARE DESFire emulator in the card for compatibility with existing access control systems. Could also attempt to implement the MIFARE Plus EV1 distance-bounding protocol.
- Add MIFARE-like transactions for payment applications, possibly using limited increment/decrement transactions for value counters. Look into anti-tearing protection for the transactions i.e. on each reset, check if the previous transaction exited prematurely leaving the card in an inconsistent state, and restore to a consistent state if necessary.

Variants

The intention is to implement everything using programmable Java Cards using the language Java, but alternatives could be considered, for example cards compatible with the Microsoft .NET smartcard API which can be programmed using C#.

The optional extensions that will be implemented, if any, could be done in any order. A judgement will be made at the time as to which ones are more useful or which could be done in a reasonable time frame.

It is not yet decided whether the system will be developed using a Windows or Linux OS.

The door controller will initially be implemented as an application running on a PC. Later, this functionality may be ported to something more suitable for mounting near a door, such as a Raspberry Pi.

Success Criteria

The metrics that will be used to judge the success of the project will be determined largely by its suitability as a hypothetical replacement to the current university smartcard system. It should not introduce needless inconvenience, and it should address the primary drawbacks of the MIFARE classic system. The following criteria will be used:

1. The time taken for the reader to be able to correctly authenticate or deny the card should be no longer than a second.

2. The access rights of a card should be revocable without having to modify the card itself.

3. The system should be sufficiently flexible to allow for unlimited doors, each with different access privileges.

4. The system should be secure - very resistant to most, or all, likely forms of attack. These include, but are not limited to, brute-force attack, card cloning, prediction of generated keystreams, unauthorised reading and writing of card memory, and tampering with door controllers. Relay attacks may be dealt with via an optional extension.

5. The estimated cost per card and per terminal of introducing the smartcard system at scale should not be higher than the typical worst-case cost of a traditional smartcard system, estimated by the smartcard consultancy CardWerk to be around £8 per card and £115 per terminal.

6. The dissertation must be planned and written.

Timetable and Milestones

Weeks 1 and 2

Do preliminary reading of information regarding MIFARE and RFID/smartcards in general, including the ISO-14443 and ISO-7816 standards, asymmetric encryption, digital signature algorithms and specific protocols. Practice using the hardware and development software to create very basic smartcard applications.

Milestones: A basic application in which the reader and smartcard successfully interact and exchange information.

Week 3

Study the existing protocols for asymmetric cryptography in access control, as well as other preliminary reading.

Milestones: Decide on a protocol to implement and other implementation strategies, have written a trivial "Hello World" app in which data can be written onto the Java Card via a self-written PC application and read by the reader.

Weeks 4 to 7

Create a basic, working implementation of the chosen protocol, such that the reader accepts a card with the correct key, otherwise the card is rejected. Test this implementation for bugs.

Milestones: Have a working prototype that can be added to later.

Weeks 8 and 9

Assess the implementation from a security point of view, considering the likely forms of attack. Find and fix security flaws.

Milestones: Have a reasonably secure implementation, and a good understanding of its shortcomings in security.

Weeks 10 and 11

Write a progress report, detailing the current implementation and the main decisions made. At the same time, plan the first part of the Dissertation pertaining to the work completed so far.

Milestones: Handed in progress report, Dissertation structure has been planned.

Week 12

Conduct tests on the smartcard system, and assess how well the current implementation satisfies the success criteria.

Milestones: Should have a reasonably complete, presentable project by this point. It should meet most, if not all, of the success criteria.

Weeks 13 to 15

If there is any work still to be done on the core project, finish it off. Address some optional extensions.

Milestones: No more work should be needed on the implementation at this point. Some optional extension work should also be completed.

Weeks 16 to 20

Finish Dissertation, creating diagrams and collating all notes made during the process. Expand notes into detailed narratives, and put them together in a structured manner.

Milestones: The project is finished, and the Dissertation has been written and is ready for submission.